基于局部划分的匿名算法研究 *

王　芳 [1]，余敦辉 [1, 2†]，张万山 [1, 2]

(1. 湖北大学 计算机与信息工程学院, 武汉 430062; 2. 湖北省教育信息化工程技术研究中心, 武汉 430062)

摘　要：针对泛化会造成数据信息损失量较大，且这缺陷会随数据维度变大而越明显的问题，提出一种基于局部划分的匿名算法。在确保 k-匿名和 l-匿名的前提下，基于敏感属性栏值约束和记录间距离将数据表横向分成若干个桶，然后对每个桶基于属性间的关联纵向分成多栏，最后对同一桶中各栏中的数据进行随机重排。实验结果表明，在处理高维数据时，与 LGAA-CP 算法相比，信息损失量减少了 47%到 183%，关联关系保留率提高了 24%~118%。与 Slicing 算法相比，信息损失量相差在 1.5%之内，关联关系保留率提高了 8.9%到 22.8%。通过分析，该算法在同时确保高维数据的隐私保护能力和数据可用性方面是有效的。

关键词：数据发布隐私保护；k-匿名；l-匿名；敏感属性栏值约束

中图分类号：TP309.2　　doi: 10.3969/j.issn.1001-3695.2018.05.0352

Anonymous algorithm based on local partition

Wang Fang[1], Yu Dunhui[1, 2†], Zhang Wanshan[1, 2]

(1. *College of Computer & Information Engineering Hubei University*, *Wuhan* 430062, *China*; 2. *Hubei Education Information Engineering & Technology Center*, *Wuhan* 430062, *China*)

Abstract: Aiming at the problem that the generalization causes a large amount of loss of data information, and this defect would become more obvious as the data dimension becomes larger, this paper proposed an anonymous algorithm based on local partitioning. To ensuring k-anonymity and l-diversity, based on the value constraint of sensitive attribute column and the distance between records, horizontally divide the data table into several buckets. And then based on the relationship between the attributes, longitudinally divide the buckets into multiple columns. Finally, randomly rearrange the columns in the same bucket. The experimental results show that when dealing with high dimensional data, compared with LGAA-CP algorithm, the loss of information is reduced by 47% to 183%. the retention rate of the relationship is increased by 24% to 118%. Compared with the Slicing algorithm, the loss of information between the two is within 1.5%. the retention rate of the association is increased by 8.9% to 22.8%. The analysis show that the algorithm is effective in ensuring both high-dimensional data privacy protection and data availability.

Key words: privacy-preserving data publishing; k-anonymity; l-diversity; sensitive attribute column value constraint

0　引言

伴随信息技术的广泛使用和大数据技术的日益普及，数据开放已经是大势所趋。由于政府、部门、企业所开放的数据中包含了大量的隐私信息，若将数据直接以原始形式发布出去，必然会造成隐私泄露[1]。在此背景下，数据发布隐私保护（privacy-preserving data publishing，PPDP）这一技术被提出。与利用数据加密和访问控制等技术防止未授权者获取数据的传统隐私保护方法不同，PPDP 的目的是尽量保持发布数据的可用性，同时防止数据使用者识别出敏感信息所对应的个体，从而实现隐私保护[2-3]。

作为该领域的研究热点，当前很多学者在这方面做了大量的研究，众多数据发布模型纷纷被提出，如 *k*-匿名模型[4,5]、*l*-匿名[6,7]和 *t*-接近性[8,9]等。但是这些研究存在的问题之一是虽然较好地实现了隐私保护，但是信息损失量大，数据可用性低。针对这一问题，很多学者在实现隐私保护的同时，开始考虑数据可用性。2012 年徐勇等人提出基于属性权重的 *k*-匿名算法 WAK-anonymity[10]，解决了重要信息损失量大的问题，但加大了其他数据的信息损失量,而且各数据间关联关系损失量很大。2012 年李珊珊等人提出基于聚类的数据敏感属性匿名保护算

收稿日期：2018-05-28；**修回日期**：2018-07-23　**基金项目**：国家重点研发计划资助项目（2016YFB0800401）；国家“973”计划资助项目（2014CB340404）；国家自然科学基金资助项目（61373037，61672387）；湖北省重大专项资助项目（2018ACA133）

作者简介：王芳（1992-），女，广西桂林人，硕士研究生，主要研究方向为大数据；余敦辉（1974-），男（通信作者），副教授，博士，主要研究方向为服务计算、大数据（yumhy@hubu.edu.cn）；张万山（1973-），男，讲师，主要研究方向为 Web 信息挖掘.

法[11]。2016 年龚奇源针对高维数据提出基于半划分的数据匿名算法[12]。2017 年廖军等人提出一种基于权重属性熵的分类匿名算法[13]。这些方法在一定程度上降低信息的损失程度，但效果不太显著。同年姜火文提出贪心聚类匿名方法[14]，实现了等价类均衡划分，进一步提高了数据可用性，但仍然无法避免高维数据信息损失量大这一问题。

为了减少信息损失量，提高数据可用性，一些新的方法被陆续提出。2012 年，Li 等人[15]提出了基于随机重排技术的 Slicing 算法思想，就如何减少信息损失量，保留属性间有用的关联关系给出了解决思路。2013 年 Li 等人[16]提出了 Slicing 算法，利用对数据表横向分桶、纵向分栏的方式，在隐私保护的前提下，一定程度上保持了属性间的关联关系。2014 年 Yang 等人[17]提出的重叠划分匿名算法，以 Slicing 算法为基础，但允许一个属性被划分到多栏，其优点是关联关系能更好地得到保留，缺点是在数据维数很高时，隐私保护能力凸显不足。2015 年 Rohilla 提出的 LDS 算法[18]，在 Slicing 的基础上只解决了连续型数据离散化的问题。2018 年 Wang 等人针对事务数据提出 T-Closeness Slicing 算法[19]，该算法使发布数据满足更严格的匿名发布 T-Closeness 模型，提高了隐私保护能力，但数据可用性有所下降。2016 年王良等人提出基于加权贝叶斯网络的隐私数据发布方法[20]，利用差分隐私保护技术提升了原始隐私发布数据集的数据精确性，但在处理高维数据时还存在一些不足。综上所述，这些方法很难同时确保具有较好的隐私保护能力和数据可用性。

为此，本文提出一种基于局部划分的 ASG-LS(association set generating- local slicing)匿名算法，首先基于泛化层次树转换形成的泛化层次格，找出敏感属性顶点集 $srset$ 和不包含敏感属性的关系属性顶点集 $rset$，然后根据 $srset$ 将数据表 T 进行横向划分成多个桶，再根据 $srset$ 与 $rset$ 将同一桶中不同属性进行纵向划分形成栏，最后把同一桶中各栏的数据进行重排。从而在实现隐私保护的同时，确保发布的数据具有更高的可用性。

1　匿名算法理论基础

假设要发布的数据表 T 已经舍去标识符属性，如姓名、身份证号等。数据表 T 包含的属性 $A=\{A_1,A_2,\cdots,A_i,\cdots,A_n,S\}$，其中 $A_i(1\le i\le n)$ 为准标识符属性，S 为敏感属性。

定义 1　泛化层次树 $TSet$。若一棵树中任意存在父子关系的节点均满足泛化关系，则称这棵树为泛化层次树。在属性 A_i 的泛化层次树中，第 j 层中第 k 个节点记做 A_{ijk}，其值域为 $D(A_{ijk})$，“<”表示属性值域间的泛化关系。若树中节点 A_{ijk} 与 $A_{ilm}(l=j+1)$ 存在父子关系，则 $D(A_{ilm})<D(A_{ijk})$ 一定成立。另外，若泛指属性 A_i 泛化层次树中的某个节点时，记做 $node_i$。

定义 2　泛化格 g_i。若一个图中任意存在边的顶点均满足泛化关系，且顶点为不同属性泛化层次树的节点集合，则称这个图为泛化格。i 个不同属性的泛化层次树构成的泛化格被记做 g_i，g_i 中的任意一个顶点 v 则是这 i 个泛化层次树中的节点 $node_1,node_2,\cdots,node_i$ 的组合。且图中的任意一条边 v_mv_k 的起始顶点 v_m 与终止顶点 v_j 满足以下条件：　$v_m<v_j$；有且仅有一个 $node_k$，使得 $v_m.node_k<v_j.node_k$，而两个顶点中的其余节点相同。

定义 3　敏感属性栏值约束. 将数据表 T 划分为桶时所遵循的约束条件。具体包括：a) 敏感属性栏应满足 l-匿名；b)敏感属性栏内的属性间可存在关联关系，但不能与该栏以外的其他属性有关联关系。

定义 4　桶 B。数据表 T 基于敏感属性栏值约束和记录间距离所做的一个横向的划分。假设一张数据表 T 被划分 n 个桶 $B_1,B_2,...B_n$，则 $\bigcup_{i=0}^{n}B_i=T$，且 $\forall k，j，B_k\cap B_j=\varnothing$，$(0\le k,j\le n)$。

定义 5　栏 C。数据表 T 中的一个桶 B 基于属性间的关联关系，可被划分为 m 栏，则 $\bigcup_{i=0}^{m}C_i=B$，且 $\forall j,k$，$C_j\cap C_k=\phi$，$(1\le j，k\le m)$。

定义 6　顶点 v 约束。记录 t 中的属性在顶点 v 中有对应的属性节点，那么该属性的取值在其对应的属性节点值域内，t $\in T$。

表 1　符号标志

符号	意义
A_i	准标志符属性
S	敏感属性
$rset$	非敏感属性关联关系顶点集合
$srset$	包含敏感属性关联关系顶点合
$TSet$	泛化层次树
$FTSet$	频繁泛化层次树，即所有节点支持度均大于支持度阈值的化层次树
$node_i$	属性 A_i 泛化层次树中的节点
g_i	泛化格
fg_i	频繁泛化格，即顶点的支持度均大于支持度阈值的泛化格
v	顶点,其中 $v\in g_i$,由 i 个泛化层次树中的节点 $node_1$，$node_2$，L，$node_i$ 组合
B	桶
w	属性权重
$SupTh$	支持度阈值
mc	栏中属性列数最大值
C	栏
t	数据表记录
$t_pA_i^c$	记录 t_p 的第 i 个分类型属性
$t_pA_i^n$	记录 t_p 的第 i 个数值型属性
$conTh$	置信度阈值

2　基于局部划分的匿名算法 ASG-LS

2.1　算法总体概述

为确保在隐私信息不被泄露的情况下，提升所发布的数据间的关联关系保持率，从而提高数据的可用性，针对发布数据

中存在属性间的关系随值域的变化而变化的情形，提出一种基于局部划分的 ASG-LS 匿名算法。该算法先后依次调用 ASG 算法和 LS 算法，其中 ASG 算法用于找出属性值域间的多维多层次关联关系，并按顶点重要度降序，将包含敏感属性的关系保存到含敏感属性关联顶点集合 $srset$，不包含敏感属性的关系保存到非敏感属性关联顶点集合 $rset$ 中；LS 算法则用于实现对数据表的划分，及匿名处理，其基本思想是基于敏感属性值约束和记录间距离将数据表横向分成若干个桶，再对每个桶基于属性间的关联纵向分成多栏，然后对同一桶中各栏中的数据进行随机重排，最终使得发布的数据在隐私保护的同时，使数据具有更高的利用价值。

2.2　属性关联关系集合生成算法 ASG

ASG 算法完成 LS 算法的预处理工作，基于 Apriori 和层次泛化树实现，目的是找到属性值域间的多维多层次关联关系，并按重要度降序保存包含关联关系的顶点，将包含敏感属性的顶点保存到敏感属性关联关系集合 $srset$ 中，而其他顶点保存到普通关联关系集合 $rset$ 中。ASG 算法如 2 所示。

该算法的主要实现过程分为四步：

a)生成频繁泛化层次树。从叶子节点开始，基于广度搜索的思想，按照从左往右，从下往上的次序遍历泛化层次树。根据节点包含的属性值域在数据表中出现的概率，依次计算每个节点的支持度，若支持度 $\geq$ 支持度阈值 $SupTh$，则保留该节点，并标记与之直接相连的节点，反之则删除该节点，继续遍历与之相连的节点。遍历过程中若节点已被标记，则直接保留节点。最终使得泛化层次树中的所有节点的支持度均 $\geq SupTh$，即得到频繁泛化层次树。

b)将频繁泛化层次树转为泛化格。频繁泛化层次树之间连接产生泛化格：选取两棵不同属性的频繁泛化层次树，两两连接生成泛化格 g_2，其顶点为两棵树中的节点的组合；若其中某两个顶点包含的节点组合中只有一组对应的属性节点存在泛化关系，而其余节点相同时，则依据泛化关系生成一条有向边。

频繁泛化层次树与频繁泛化格 fg_{i-1} 连接产生泛化格 g_i：若频繁泛化格 fg_{i-1} 没有包含该频繁泛化层次树，则进行连接，并采用上述方法生成泛化格 g_i。否则不进行连接。

c)生成敏感及非敏感属性关联关系集合。从没有边指向的顶点开始，基于广度搜索的思想，按层次，从下往上的次序遍历计算泛化格 g_i 中各顶点的支持度与置信度。若两者均大于等于设定阈值，则保存该顶点，并根据其是否包含敏感属性，分别保存到敏感属性关联关系集合 $srset$ 或非敏感属性关联关系集合 $rset$ 中，然后标记与之直接相连的顶点；反之则删除该顶点。遍历过程中若顶点已被标记，则直接保留顶点。最终关联关系集合中保存了泛化格中具有关联关系、顶点包含的节点层次相对最大和各顶点中对应节点值域相交为空的顶点；泛化格变成频繁泛化格。

d)实现顶点的重要度排序。为保留更多有用的关联关系，在本文后续 LS 算法中优先按包含重要关联关系的顶点进行划分，因此需要将关联关系集合 $srset$ 中各顶点按重要度降序排序。

顶点 v 的重要度 $value(v)$ 是评价一个顶点中包含的属性间关联关系是否重要的指标。它由顶点中所包含的所有节点的层次和顶点的置信度共同决定，为此，首先利用式(1)计算节点 $node_i$ 归一化后的层次值。其中，lev_{node_i} 为节点 $node_i$ 在泛化层次树中的层次，$lev_{\max}$ 为泛化层次树的层次最大值。

$$lev(node_i)=\frac{lev_{node_i}-lev_{root}}{lev_{max}-lev_{root}},\tag{1}$$

然后，再利用式(2)计算顶点 v 的重要度 $value(v)$。其中，w_i 表示数据表 T 中属性 A_i 的权重，n 为顶点 v 中包含节点的个数，$conf(v)$ 为顶点 v 的置信度。

$$value(v)=\sum_{i=1}^{n}w_i lev(node_i)+conf(v),\tag{2}$$

算法 1　ASG 算法

输入：数据表 T，属性泛化层次树 $TSet$，置信度阈值 $confTh$，支持度阈值 $supTh$，栏中属性列最大值 mc，属性权重 w；

输出：含敏感属性关联顶点集合 $srset$，非敏感属性关联顶点集合 $rset$。

1. 遍历 $TSet$，计算树中各节点支持度，剪枝支持度小于支持度阈值 $supTh$ 的节点，找出频繁泛化层次树，并保存到频繁泛化层次树集合 $FTSet$；
2. $FTSet$ 中所有频繁泛化层次树，两两连接生成泛化格，并加入泛化格集合中；
3. for (每一个 $g_i \in$ 泛化格集合)
4. 所有没有边指向的顶点，按层次把顶点入队；
5. while (队不为空)
6. if 顶点没有被标记
7. 计算泛化格 g_i 中各顶点的置信度 $conf$ 和支持度 $support$；
8. if $support \geq supTh$ 且 $conf \geq conTh$
9. 按该顶点是否包含敏感属性，分别保存到敏感属性关联关系集合 $srset$ 和非敏感属性关联关系集合 $rset$；

else

10. 删除该顶点，与之直接相连的顶点入队；

end if

end if

end while

11. 泛化格 g 加入频繁泛化格集合 fgset；

end for

12. fgset 中各泛化格 g_i 与 $FTSet$ 中各频繁泛化层次树连接，并加入新的泛化格集合中；
13. 转步骤 3 迭代，直到泛化格中顶点包含的节点数 $\geq mc$；
14. 基于顶点包含的各属性权重 w，计算各顶点重要度 $value(v)$，并按顶点重要度降序排序 $srset$ 中的各顶点；
15. return $srset$, $rset$；

2.3　基于局部划分的 LS 算法

LS 算法基于 Slicing 思想和关联关系集合，采用局部划分

数据表的匿名方法。其基本思想是先横向划分数据表，将整个数据表划分成多个桶，然后分别对每个桶进行纵向划分，将一个桶划分为多栏，其结果是同一属性在不同桶中不一定在同一栏。从而确保既减少信息损失量，又减少数据间关系的损失量。LS 算法如算法 2 所示。

该算法的主要实现过程分四步：

a) 对数据表横向分桶

为减少数据的信息损失量，对数据表进行横向划分桶。划分时每次从 $srset$ 中取出一个重要度值最大的顶点 v，然后从数据表 T 中随机取出一条满足顶点 v 值约束的记录 t，加入桶 B，进而从数据表 T 中挑选 k 条满足敏感属性栏值约束和距离最近约束的记录到桶 B。

为挑选出满足距离最近约束的记录，必须计算数据表中任意一条记录与记录 t 的距离，其值采用两条记录的属性间的距离来度量。属性间的距离分为数值型属性间的距离与分类型属性间的距离，记录 t_p 的第 i 个数值型属性与记录 t_q 的第 i 个数值型属性间的距离为 $dist(t_pA_i^n,t_qA_i^n)$，记录 t_p 的第 i 个分类型属性与记录 t_q 第 i 个分类型属性间的距离为 $dist(t_pA_i^c,t_qA_i^c)$。

数值型准标识符属性间的距离 $dist(t_pA_i^n,t_qA_i^n)$：

$$dist(t_pA_i^n,t_qA_i^n)=\frac{\left|D\left(t_pA_i^n\right)-D\left(t_qA_i^n\right)\right|}{max\left(TA_i^n\right)-\min\left(TA_i^n\right)}, \quad (3)$$

$D\left(t_pA_i^n\right)$ 表示元组 t_p 的第 i 个数值型准标识符的取值，$\left|D\left(t_pA_i^n\right)-D\left(t_qA_i^n\right)\right|$ 为取两个属性值之差的绝对值，$max\left(TA_i^n\right)$ 表示数据表 T 第 i 个数值型标识符的最大值，$\min\left(TA_i^n\right)$ 表示数据表 T 第 i 个数值型标准符的最小值。

分类型准标识符属性间的距离计算需要引入泛化层次树作为度量依据，参照泛化层次树首先计算元组 t_p 第 i 个分类型属性 $t_pA_i^c$ 与元组 t_q 第 i 个分类型属性 $t_qA_i^c$ 到共同最小祖先节点的距离，记为 $len(t_pA_i^c,\ t_qA_i^c)$；然后计算 $t_pA_i^c$，$t_qA_i^c$ 各自到根节点的距离 $len(t_pA_i^c,root)$；最后计算分类型准标识符属性间的距离 $dist(t_pA_i^c,t_qA_i^c)$，公式如下。

$$dist(t_pA_i^c,t_qA_i^c)=\frac{len(t_pA_i^c,t_qA_i^c)}{len(t_pA_i^c,root)+len(t_qA_i^c,root)}, \quad (4)$$

假设一个记录有 d_1 个数值型准标识符，d_2 个分类型准标识符，记录间的距离为 $dist(t_p,t_q)$。

$$dist(t_p,t_q)=\sum_{i=1}^{d_1}dist(t_pA_i^n,t_qA_i^n)+\sum_{i=1}^{d_2}dist(t_pA_i^c,t_qA_i^c), \quad (5)$$

b) 栏划分

为减少数据间关系的损失量，将每个桶纵向划分成多个栏。其思想是针对每个桶，首先将包含敏感属性的顶点 v 所包含的节点集对应的属性划分到一栏。再遍历非敏感属性关联关系集合 $rset$ 中每一个顶点 v_i，计算桶中满足顶点 v_i 约束的记录数。并按记录数降序排序 $rset$ 后，按照从前往后的次序依次取出顶点 v_i，找出 v_i 中所包含的各节点所对应的桶中属性，并去除其中已经被分配的属性，若剩余属性数量大于或等于设定的栏中最小属性数量，则将其划分为一栏，否则放弃该划分。最后，若桶中剩余的未被纵向划分的属性列数大于栏中最大属性数量，则对属性进行均匀分栏，否则将剩余属性直接作为一栏。

c) 数据重排

假设桶中的属性共划分为 c 栏，则对桶中任意 c-1 栏进行随机重排；重排后，若有记录仍处于原来位置，则将其与桶中其他记录随机置换。因此同栏中高关联属性间的关联关系保持不变，破坏的是不同栏之间的关联关系，从而保持了高关联属性间的关联关系。

d) 栏概化

为使数据表 T 满足 k- 匿名性。检查桶中某属性值在数据表 T 中出现的概率，若低于阈值 f，则需对其所在的栏进行该属性列的概化；列概化后，计算栏中不重复的记录数等于 k 的栏数量，若栏数<2，则对整个桶进行概化。

算法 2　LS 算法

输入：数据表 T，$srset$，$rset$，k- 匿名性参数 k，l- 多样性参数 l，栏中属性最大数量 mc，栏中最小属性数量 nc

输出：匿名后的数据表 T^*

1.　while（$srset$ 不为空且 T 中存在未处理的记录）
2.　从 $srset$ 中按从前往后的次序取出顶点 v；
3.　while (满足顶点 v 约束的记录数>0)
4.　从数据表 T 中随机取出一条满足顶点 v 约束的记录 t，加入桶 B；
5.　while (桶 B 中的记录数< k)
6.　取出与 t 距离最近且满足敏感属性栏值约束的记录，加入桶 B 中；
7.　end while
8.　将顶点 v 包含的节点集对应的属性划为一 栏；
9.　for (每一个顶点 $v_i \in rset$)
10.　计算桶中满足顶点 v_i 约束的记录数 nt；
11.　end for
12.　按记录数 nt 降序排序 $rset$ 中各个顶点；
13.　for (按从前往后的次序从 $rset$ 中取出顶点 v_i)
14.　找出 v_i 包含的各节点在桶 B 中对应的属性，并去除其中已经被分配到其他栏的属性；
15.　若剩余属性列数 $>nc$，将其划分为一栏；否则放弃该划分；
16.　end for
17.　若桶中剩余的未划分栏的属性列数 $>mc$，对属性进行均匀分栏；否则将剩余属性直接作为一栏；
18.　若桶 B 总共被划分 n 栏，则对其中 n-1 栏进行随机重排；
19.　重排后，若存在记录仍处于原来位置，则将其与桶中其他记录随机置换；
20.　检查桶中某属性值在数据表 T 中出现的概率；
21.　若低于阈值 f，则需对其所在的栏进行该属性列的概化；
22.　列概化后，计算栏中不重复的记录数等于 k 的栏数量，若栏数<2，则对整个桶进行泛化。

23.　end while

24.　end while

25.　return T^*

2.4　时间复杂性分析

ASG-LS 匿名算法由两个子算法 ASG 和 LS 组成，其中，ASG 算法生成频繁泛化层次树，以及将频繁泛化层次树转为泛化格的时间复杂度均为 $O(n^2)$；另外两个步骤时间复杂度均为 $O(n)$；所以子算法 ASG 的时间复杂度为 $O(n^2)$。LS 算法的时间复杂度只与 $srset$ 中的顶点数、$rset$ 中的顶点数、以及 k-匿名性参数 k 以及表 T 中的记录数 n 有关。顶点数与 k 值远小于 n，因此时间复杂度为 $O(n^2)$。所以 ASG-LS 算法总的时间复杂度为 $O(n^2)$。

3　算法比较与实验分析

3.1　算法比较

将本文提出的 ASG-LS 算法与泛化和 Slicing 两类方法进行比较。根据某家医院的病人诊断记录原始表，如表 4 所示，泛化首先把距离较近的记录划分到同一个等价类，然后概括等价类中各属性所有列中的值，如表 5 所示。Slicing 横向划分将距离较近的记录划分到同一个桶中，纵向全局划分将高关联关系的数据划分到同一栏中，最后随机重排同栏中的数据，如表 6 所示。

表 4　病人诊断记录原始表

Name	Age	sex	District	Disease
Tom	22	M	Hawaii	bronchitis
Lily	27	F	California	breastcancer
Hebe	35	F	California	pnenmonia
John	49	M	Connecticut	flu
Ella	59	M	Connecticut	flu
Selina	60	F	Mississippi	breastcancer
Bob	65	M	Washington	gastritis
Lau	70	F	Alaska	breastcancer

对表 4 中数据进行泛化处理后，结果如表 5 所示，从表中可看出，该结果满足 4-匿名性和 3-多样性，使发布的数据表满足了一定程度的隐私保护要求，但数据过度泛化导致数据的信息损失量过大，数据可用性很低。

表 5　泛化后的病人诊断记录表

age	sex	district	zipcode	disease
[21-27]	*	America	47*	flu
[21-27]	*	America	47*	endemictyphus
[21-27]	*	America	47*	flu
[21-27]	*	America	47*	dyspepsia
[35,50]	*	America	47*	pnenmonia
[35,50]	*	America	47*	bronchitis
[35,50]	*	America	47*	breastcancer
[35,50]	*	America	47*	breastcancer

采用 Slicing 算法处理后，结果如表 6 所示，从结果可看出，该算法将敏感属性和与其有关的属性划分为一栏，其他属性分为一栏，并通过栏中数据进行置换来实现隐私保护。其结果满足 3-多样性和 4-匿名性，达到了一定程度的隐私保护。和泛化方法相比，Slicing 算法不仅进行横向划分，还进行了纵向划分，从而提高了数据可用性。但其缺点是属性间的关系随值域的变化而变化，会损失部分属性间的关联关系。

表 6　Slicing 算法处理后病人记录诊断记录表

age, sex	district,zipcode,disease
22, M	California, 47905, endemictyphus
25, F	Connecticut, 47302, flu
26, M	Bloomington, 47490, dyspepsia
27, F	Connecticut, 47302, flu
35, M	Alaska, 47304, breastcancer
40, F	Washington, 47301, pnenmonia
49, M	Mississippi, 47904, breastcancer
50, F	Washington, 47301, bronchitis

采用本文的 ASG-LS 算法处理后如表 7 所示。整个表被分为两个桶，每个桶各自被划分为两栏。从结果不难发现，其满足 4-匿名性与 3-多样性，能较好地满足隐私保护要求。与 Slicing 算法划分后同一属性在不同桶中必在同一栏不同，由 ASG-LS 算法纵向划分后，利用在不同桶中，同一属性被划分到不同栏的方法，更多地保留了属性之间的关联关系，从而大大提高了数据的可用性。

表 7　ASG-LS 算法处理后的病人记录诊断记录表

Age,Sex	District,Zipcode,Disease
22, M	Bloomington,47490,dyspepsa
25, F	Connecticut,47302,flu
26, M	California,47905,endemictys
27, F	Connecticut, 47302,flu
Age,District,Zipcode	Sex, Disease
35,Washington,47490	F, breastcancer
40,Connecticut,47302	F, breastcancer
49, Alaska, 47304	M, bronchitis
50, Mississippi, 47904	M, pnenmonia

3.2　实验分析

从信息损失量[13]和数据可用性两个方面对本文提出的 ASG-LS 算法的性能进行验证。GAA-CP[14]是信息损失量较小的泛化处理算法，其具有一定优秀泛化算法的代表性。但 GAA-CP 只满足 K-匿名，为使能更好的对比将其满足 *l*-匿名，记为 LGAA-CP。Slicing[16]算法中的 k 值为桶中记录数的最小值。

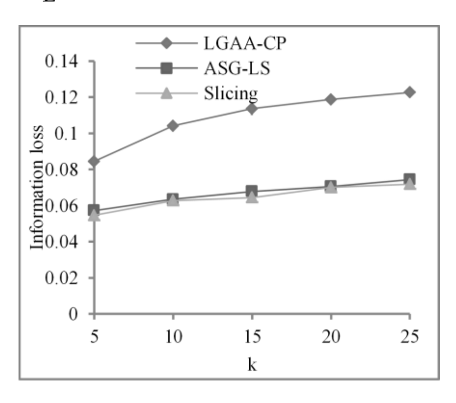
实验数据集来源于隐私保护领域广泛使用的 Adult 数据集。在去除该数据集中具有缺失属性值的记录后，最终实验基于一个包含 30162 个元组的数据集完成。实验环境为：Intel(R) Core™ i5-2450M CPU @2.50GHz；4.00 GB 内存；LITEON T9 (256 GB) 主硬盘；Windows 10 专业版 64 位操作系统；MySQL 数据库系

统；IntelliJ IDEA2017.2.5 开发环境；jdk8 运行环境。算法采用 Java 语言实现。

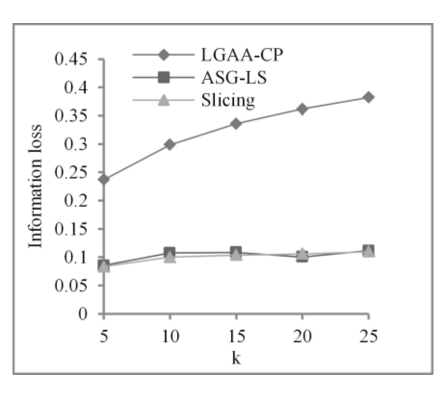
3.2.1 信息损失分析

为分析数据信息损失量随数据集维度 $|T_m|$、匿名参数 k 值改变而改变的规律，分别采用 ASG-LS 算法、LGAA-CP 算法和 Slicing 算法进行如下两组实验。

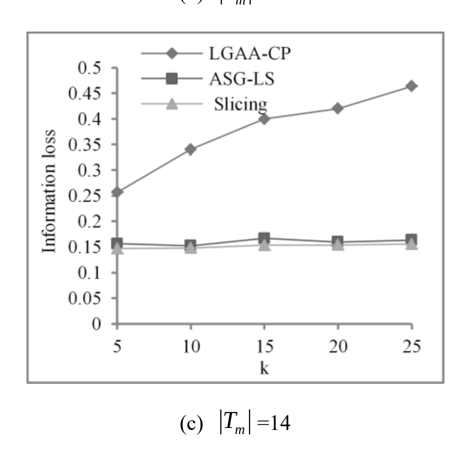
实验 1　当数据维度 $|T_m|$ 分别为 5、10、14 时，验证三种算法中匿名参数 k 值的变化对信息损失量大小的影响。结果如图 1(a)(b)(c)所示。其中栏中属性最大数量 mc =4，栏中最小属性数量 $nc=2$，置信度阈值 $confTh=0.8$，支持度阈值 $supTh=0.2$，l-多样性参数 L =3。



(a)　$|T_m|$ =5



(b)　$|T_m|$ =10

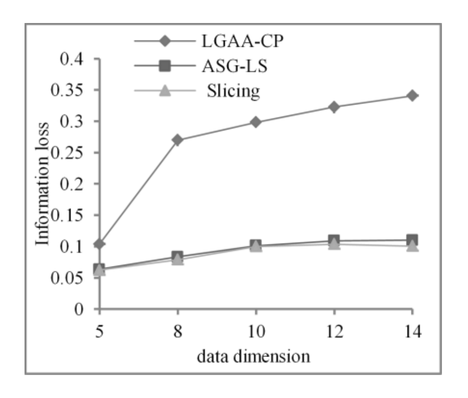


(c)　$|T_m|$ =14

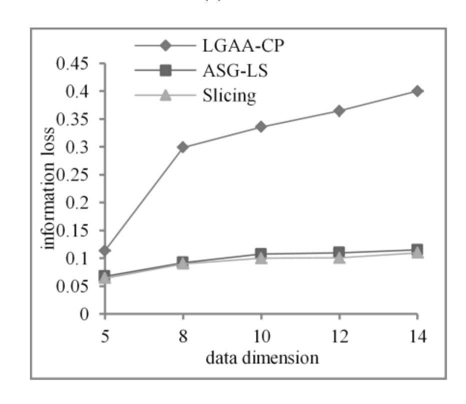
图 1　匿名参数 k 值的变化对信息损失量影响

实验结果表明，当数据维度相同时，随着 k 值的增大，三者的数据信息损失量都增大，但 LGAA-CP 的信息损失量显著增大。Slicing 和 ASG-LS 的信息损失量增加很少，且两者基本一致，因为 Slicing 和 ASG-LS 都是进行的局部的泛化处理。

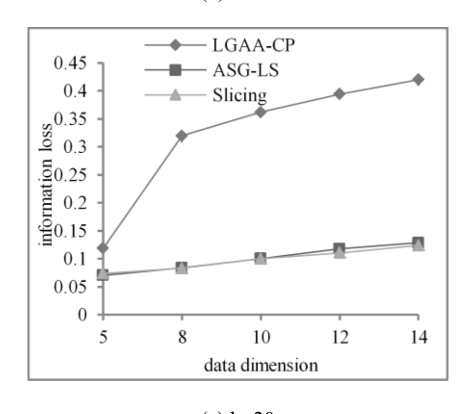
实验 2　当匿名参数 k 分别为 10、15、20 时，验证三种算法中数据维度 $|T_m|$ 的变化对信息损失量大小的影响。结果如图 2(a)(b)(c)所示。其中：栏中属性最大数量 mc =3，栏中最小属性数量 $nc=2$，置信度阈值 $confTh=0.8$，支持度阈值 $supTh=0.2$，L-多样性参数 L =3。



(a) k=10



(b) k=15



(c) k=20

图 2　数据维度 $|T_m|$ 变化对信息损失量影响

实验结果表明，当匿名参数 k 值相同时，随数据维度 $|T_m|$ 的增大，三者的数据信息损失量都增大。但 LGAA-CP 的信息损失量显著增大，其中当 $|T_m|$ 取值在 5 到 8 之间时增长最快；而同样因为 Slicing 和 ASG-LS 进行的都是局部泛化处理，其信息损失量较小，且增速较慢。

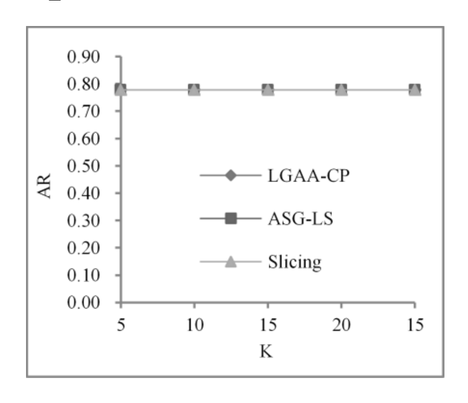
3.2.2 数据可用性分析

关联关系保持率 AR 是数据匿名处理后保留的关联规则数与原始数据关联规则数的比值，用于衡量匿名方法在保留属性间有用关联关系效果。本文采用 Weka Apriori 挖掘关联规则。

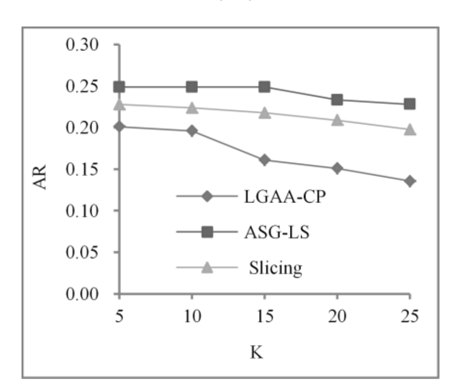
为了分析匿名后数据的关联关系保持率随数据集维度|Tm|、

k-匿名对应 *k* 值改变而改变的规律，分别采用 ASG-LS 算法、LGAA-CP 和 Slicing 算法进行如下两组实验。

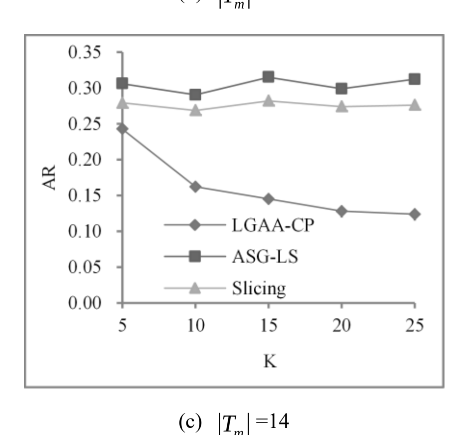
实验 3 当数据维度 $|T_m|$ 分别为 5、10、14 时，验证三种算法中匿名参数 k 值的变化对关联关系保持率的影响。结果如图 3(a)~(c)所示。其中：栏中属性最大数量 mc =4，栏中最小属性数量 $nc=2$，置信度阈值 $confTh=0.8$，支持度阈值 $supTh=0.2$，*l*-多样性参数 L =3。



(a) $|T_m|$=5



(b) $|T_m|$=10

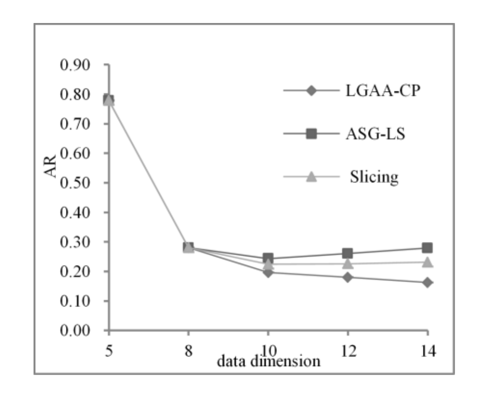


(c) $|T_m|$=14

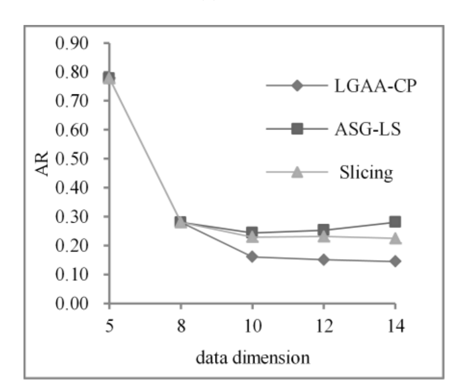
图 3 匿名参数 k 值的变化对关联关系保持率影响

实验结果表明，当数据维数 $|T_m|$=5 时，三个算法的关联关系保持率是一样的，原因是当维度较小时，原始数据具有的关联规则比较少，且很多数据自身特性已满足 k-匿名性，所以对算法的依赖性不强。而当数据维度为 10 和 14 时，随 k 值增加，LGAA-CP 算法关联关系保持率大幅度减少，ASG-LS 算法和 Slicing 算法关联关系保持率略有下降，但 ASG-LS 始终保持在 Slicing 之上。因为 ASG-LS 和 Slicing 都是进行的局部泛化处理，且 ASG-LS 是进行的是局部划分。

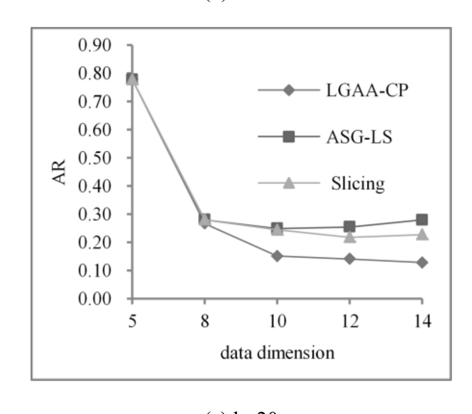
实验 4 当匿名参数 k 分别为 10、15、20 时，验证三种算法中数据维度 $|T_m|$ 的变化对关联关系保持率的影响。结果如图 4(a)(b)(c)所示。其中：栏中属性最大数量 mc =3，栏中最小属性数量 $nc=2$，置信度阈值 $confTh=0.8$，支持度阈值 $supTh=0.2$，*l*-多样性参数 L =3。



(a) k=10



(b) k=15



(c) k=20

图 4 数据维度 $|T_m|$ 变化对关联关系保持率影响

实验结果表明,三种算法均在数据维度 $|T_m|$ 取值在 5 到 8 之间时关联关系保持率急速下降，原因是数据维度从 5 增加到 8 时,原始数据中具有的关联关系数量显著增加。而当数据维度 $|T_m|$ 从 8 增加到 14 时，三种算法随数据维度增加，关联关系保持率平缓减少,但由于 ASG-LS 采取的是小局部的泛化处理和局部划分，最终导致 ASG-LS 的关联关系保持率大于 Slicing 和 LGAA-CP。

4 结束语

为了确保在隐私信息不被泄露的情况下，提高数据匿名后

的可用性，本文提出了一种基于局部划分的匿名算法研究。该算法借助 Slicing 的思想，在确保 K-匿名性和 L-匿名的前提下，基于敏感属性栏值约束和记录间距离将数据表横向分成若干个桶，然后对每个桶基于属性间的关联纵向分成多栏，最后对同一桶中各栏中的数据进行随机重排。实验结果表明，该方法在减少信息损失量和提高关联关系保留率方面具有较高的合理性和有效性。

参考文献：

[1] 武毅，王丹，蒋宗礼. 基于事务型 k-anonymity 的动态集值属性数据重发布隐私保护方法 [J]. 计算机研究与发展, 2013, 50 (S1): 248-256. (Wu Yi, Wang Dan, Jiang Zongli. Privacy preserving in re-publication of dynamic set-valued data based on transactional k-anonymity [J]. Journal of Computer Research and Development, 2013, 50 (s1): 248-256.)

[2] 冯登国，张敏，李昊. 大数据安全与隐私保护 [J]. 计算机学报, 2014, 37 (1): 246-258. (Feng Dengguo, Zhang Ming, Li Hao. Big data security and privacy protection [J]. Chinese Journal of Computers, 2014, 37 (1): 246-258.)

[3] 刘雅辉，张铁赢，靳小龙，等. 大数据时代的个人隐私保护 [J]. 计算机研究与发展, 2015, 52 (1): 229-247. (Liu Yahui, Zhnag Tieying, Jin Xiaolong. Personal privacy protection in the era of big data [J]. Journal of Computer Research and Development, 2015, 52 (1): 229-247.)

[4] Latanya S. k-anonymity: a model for protecting privacy [J]. International Journal on Uncertainty, Fuzziness and Knowledge-Based Systems, 2002, 10 (5): 557–570.

[5] 董芳菲. 基于 k-匿名的隐私保护算法研究 [D]. 兰州：西北师范大学, 2015. (Dong Fangfei. Research on privacy protection algorithm based on k-anonymity [D]. Lanzhou: Northwest Normal University, 2015.)

[6] Machanavajjhala A, Kifer D, Gehrke J. l-diversity: privacy beyond k-anonymity [C]// Proc of International Conference on Data Engineering. 2006: 24.

[7] Yang Gaoming, Li Jingzhao, Zhang Shunxiang, *et al.* An enhanced l-diversity privacy preservation [C]// Proc of International Conference on Fuzzy Systems and Knowledge Discovery. 2014: 1115-1120.

[8] Li Ninghui, Li Tiancheng, Venkatasubramanian S. t-closeness: privacy beyond k-anonymity and l-diversity [C]// Proc of International Conference on Data Engineering. IEEE, 2007: 106-115.

[9] 张健沛，谢静，杨静，等. 基于敏感属性值语义桶分组的 t-closeness 隐私模型 [J]. 计算机研究与发展, 2014, 51 (1): 126-137. (Zhang Jianpei, Xie Jing, Yang Jing, *et al.* A T-Closeness privacy model based on sensitive attribute values semantics bucketization [J]. Journal of Computer Research and Development, 2014, 51 (1): 126-137.)

[10] 徐勇，秦小麟，杨一涛，等. 一种考虑属性权重的隐私保护数据发布方法 [J]. 计算机研究与发展, 2012, 49 (5): 913-924. (Xu Yong, Qin Xiaolin, Yang Yitao, *et al.* A weight-aware approach to privacy preserving publishing data set [J]. Journal of Computer Research and Development, 2012, 49 (5): 913-924.)

[11] 李珊珊，朱玉全，陈耿. 基于聚类的数据敏感属性匿名保护算法 [J]. 计算机应用研究, 2012, 29 (2): 469-471. (Li Shanshan, Zhu Yuquan, Chen Geng. Clustering-based algorithm for data sensitive attributes anonymous protection [J]. Application Research of Computers, 2012, 29 (2): 469-471.)

[12] 龚奇源. 面向数据发布的数据匿名技术研究 [D]. 南京：东南大学, 2016. (Gong Qiyuan. Research on data anonymous technology for data publishing [D]. Nanjing: Southeast University, 2016.)

[13] 廖军，蒋朝惠，郭春，等. 一种基于权重属性熵的分类匿名算法 [J]. 计算机科学, 2017, 44 (7): 42-46. (Liao Jun, Jian Chaohui, Guo Chun, *et al.* Classification anonymity algorithm based on weight attributes entropy [J]. Computer Science, 2017, 44 (7): 42-46.)

[14] 姜火文，曾国荪，马海英. 面向表数据发布隐私保护的贪心聚类匿名方法 [J]. 软件学报, 2017, 28 (2): 341-351. (Jiang Huowen, Zeng Guosun, Ma Haiying. Greedy Clustering-Anonymity Method for Privacy Preservation of Table Data-publing [J]. Journal of Software, 2017, 28 (2): 341-351.)

[15] Li Tiangcheng, Li Ninghui, Zhang Jian, *et al.* Slicing: A New Approach for Privacy Preserving Data Publishing [J]. IEEE Trans on Knowledge & Data Engineering, 2012, 24 (3): 561-574.

[16] Li Tiangcheng, Li Ninghui, Zhang Jian, *et al.* Slicing: a new approach to privacy preserving data publishing [J]. International Journal of Computer Trends & Technology, 2013, 4 (8): 64-78.

[17] Yang Jing, Liu Ziyun, Yang Yue, *et al.* A data anonymous method based on overlapping slicing [C]// Proc of International Conference on Computer Supported Cooperative Work in Design. 2014: 124-128.

[18] Bhardwaj M, Rohilla S. Privacy preserving data publishing through slicing [J]. American Journal of Networks and Communications, 2015, 4 (3-1): 45-53.

[19] Wang Mingzheng, Jiang Zhengrui, Zhang Yu, *et al.* t-closeness slicing: a new privacy preserving approach for transactional data publishing [J]. Social Science Electronic Publishing, 2018, 29 (7): 50-63.

[20] 王良，王伟平，孟丹. 基于加权贝叶斯网络的隐私数据发布方法 [J]. 计算机研究与发展, 2016, 53 (10): 2343-2353. (Wang Liang, Wang Weiping, Meng Dan. Privacy preserving data publishing via weighted bayessian networks [J]. Journal of Computer Research and Development, 2016, 53 (10): 2343-2353.)